Using Molecular Modeling to Engineer Proteins with Novel Functions

By: Vlad Codrea

Abstract:

The RCSB Protein Data Bank currently stores over 30,000 X-ray crystal structures, information that has proven invaluable in various studies that have been undertaken to analyze these proteins. My goal is to apply molecular modeling techniques in order to add new functionality to enzymes whose 3D structures are available. The standard processes for modifying the functions of proteins have traditionally remained centered on experimental work, with a heavy reliance on directed evolution. One such approach has been used by Professor Peter Schultz from the Scripps Research Institute to introduce the unnatural amino acid 3-(2-naphthyl)-L-alanine into the amber stop codon of *E. coli*. This approach uses positive and negative selection to wean the cells into taking up 3-(2-naphthyl)-L-alanine. In contrast, I have used a computational method to predict mutations in aminoacyl-tRNA synthetases, the enzymes responsible for joining amino acids with their respective tRNAs, that will change the affinities of those proteins. My goal is to arrive at aminoacyl-tRNA synthetases that are capable of incorporating analogs of arginine, cysteine, phenylalanine, and tryptophan instead of the natural residues. As a follow-up, I have prepared a mutant tyrosyl-tRNA synthetase and tRNA to test how well an analog of tyrosine is incorporated into proteins.

I have used an in-silico protein modeling framework developed by Professor Homme Hellinga from Duke University to tackle another outstanding problem in

molecular biology: how to recreate the high affinity of the streptavidin/biotin complex. The specificity of streptavidin is changed so that it binds preferentially to biotin derivatives. Streptavidin is a tetrameric protein, similar to the avidin protein found in egg whites, but made by the *Streptomyces avidinii* bacteria. Streptavidin binds tightly to the vitamin D-biotin, and forms one of the strongest naturally-occurring non-covalent interactions between a protein and an organic ligand, with a dissociation constant (Kd) on the magnitude of 10^{-15} M.

Because of this strong binding, the streptavidin/biotin combination has been used extensively in molecular and bioengineering studies that require the joining of different molecules that would not normally come together. One of the current limitations with using this combination is that it is only possible to specifically bring together two compounds (namely the compound attached to biotin and the compound attached to streptavidin) at any one step of an assay. The aim is to engineer orthologous pairs of mutant streptavidins and biotin analogs, each of which can be covalently attached to a distinct molecular payload depending on the end-user's intended application. The members of one orthologous pair will not cross-react with the complementary member of another orthologous pair – in other words, a mutant streptavidin should have a relatively poor binding affinity to biotin. This provides an element of selectivity to parallel reactions performed in the same environment.

Introduction:

Part I: Incorporation of amino acid analogs into proteins

Aminoacyl-tRNA synthetases (AARS for short) are essential enzymes in the process of gene expression and catalyze the formation of an ester bond between the carboxyl group of the amino acid and the 3' OH of the transfer RNA (tRNA). Once this covalent bond has been formed, the charged tRNA and will carry its amino acid payload to the ribosome.[1] The ribosome is responsible for catalyzing the initiation, elongation, and termination of protein synthesis. It is the hub for translation of proteins, which are generated through the concatenation of residues, one at a time, during the elongation phase. The ribosome brings together the messenger RNA (mRNA) and the appropriate charged tRNA by facilitating recognition of the codon on the mRNA by the anti-codon loop on the tRNA.[2]

The codon consists of three bases on the mRNA that code for a particular amino acid, while the anticodon consists of thee bases on the an arm of the tRNA molecule that can hydrogen bond with the codon. The hydrogen bonding is not exact in that it allows a certain degree of wobbling in the last base of a codon.[3] This is achieved by the inclusion of non-standard bases such as inosine and pseudouridine in the anticodons of tRNAs, since these bases can hydrogen bond with multiple other bases. The end result is that the same charged tRNA can recognize more than one codon, and consequently that multiple codons can designate the same amino acid.[4]

Despite the fact that there are 3^4 or 81 possible codons (three positions and 4 bases at each position), only 20 amino acids are used in constructing the vast majority of known proteins. Not all codons code for amino acids: stop codons signify the end of

protein translation and tell the ribosome to cleave the nascent peptide.[5] It is possible to use these stop codons for novel purposes, such as changing their function to that of coding for an unnatural amino acid.[6] Unnatural amino acids share the basic structure of the approximately twenty alpha-amino acids that are found in living organisms. They all contain an amino group and a carboxylate group that are attached to an alpha carbon.[7] Also attached to this alpha carbon is a side chain, conventionally represented by the letter R. The side chains vary in length and composition of atoms, and give each amino acid its unique properties. [8]

The functions that proteins can perform are typically limited by the chemical properties of the amino acids that they are composed of, although exceptions to this rule occur when proteins are granted new moieties through covalent post-translational modifications such as acetylation and glycosylation. Another common way in which proteins perform complex chemistry reactions is by coordinating with metal ions and sequestering them within the active site. One drawback is that many post-translational modifications, such as phosphorylation, are not permanent and can be readily reversed. Furthermore, determining exactly where on the protein surface to add the functional group is a complex task which hundreds of enzymes have evolved to perform and regulate.[9-11]

In order to allow maximum specificity in dictating the number and location of modifications made to a protein, it is best to introduce these modifications as the protein is being synthesized: this way, the new functional groups are an inherent part of the

nascent peptide and their location is determined by the position of the residue to which they are attached. A lab-based directed evolution approach to incorporating unnatural amino acids into specific locations of a protein has been developed by Peter Schultz from the Scripps Research Institute. This approach allows specific codons to code for the unnatural amino acid that carries on it the new functional group.[12, 13]

A codon consists of three nucleotides that lie in the open reading frame (ORF) of an mRNA strand and that can have one of two different roles in protein translation: 1) to tell the protein synthesizing machinery which amino acid should go at that position; and 2) to tell the ribosome when to stop synthesizing the current peptide. Codons are read from the 5’ to 3’ direction and their order determines the order in which amino acid residues are added from the N-terminus end to C-terminus end. There are three types of stop codons: those designated as *amber* consist of a UAG sequence on the mRNA, those designated as *opal* consist of UGA, and those designated as *ochre* consist of UAA. tRNAs are responsible for bringing the right amino acid to a particular codon, and they have two opposite ends that help them perform this function.[2, 5, 14]

In three dimensions, tRNAs are roughly shaped like the letter L, with both 5’ and 3’ ends pointing to the side and the anticodon loop pointing down. The amino acid is found covalently bound to the CCA 3' end of the tRNA, while the anticodon loop contains three nucleotides that are antiparallel and complimentary to the nucleotides of a particular codon on the mRNA. Changing the specificity of a tRNA so that it recognizes a different codon is as easy as mutating the anticodon region of the tRNA gene. Normally,

tRNAs whose anticodons recognize stop codons don't exist, but it is easy to mutate a tRNA's anticodon loop to recognize one of the stop codons. Such tRNAs are called suppressor tRNAs because they suppress termination of translation; they instead cause the insertion of their payload into the peptide and the continuation of translation of the remaining portions of mRNA.[6]

The challenging part of incorporating unnatural amino acids is creating an aminoacyl-tRNA synthetase that recognizes the unnatural amino acid and that joins the amino acid to a tRNA that recognizes a stop codon. The cleanest implementation of this plan is to bring in a pair consisting of one synthetase and its tRNA from a foreign organism into the organism that will express the proteins containing unnatural amino acids. This ensures that two key requirements are met: 1) suppressor tRNAs are not charged by any of the host organism's aminoacyl-tRNA synthetases; and 2) the mutated synthetase will not charge any of the host organism's native tRNAs. Point 1 is needed because the desired outcome is not to insert a natural amino acid into the stop codon. Point 2 is needed because mutating the organism's native synthetase to recognize an unnatural amino acid will lead to a loss in the organism's ability to incorporate the natural amino acid at its canonical codon.[15]

The foreign synthetase and tRNA can therefore form an orthogonal pair, ensuring that only the amino acid added by the foreign synthetase will be inserted at the codon recognized by the foreign tRNA. Previous studies have found that ribosomes in *E. coli* will readily accept tRNAs derived from *Methanococcus jannaschii.* Similarly, ribosomes

do not verify whether the R group of an amino acid matches the tRNA that is carrying it, so no problems are expected in the final step of protein translation. [16, 17]

Schultz et. al. have evolved, through experimental techniques, a mutant tyrosyl-tRNA synthetase that aminoacylates 3-(2-naphthyl)-L-alanine with an amber suppressor tRNA.[18] That study used the tyrosyl-tRNA synthetase from the same organism as in this experiment (*Methanococcus jannaschii).* The previous study chose five residues that were at most 7 Angstroms away from the para position of the aryl ring of the tyrosine ligand. These residues (Tyr 32, Asp 158, Ile 159, Leu 162, and Ala 167) were first mutated to alanines and then subjected to random mutagenesis by PCR. The library of cells with these mutations were grown in minimal media containing no tyrosine and chloramphenicol. They were also transformed with the chloramphenicol acetyltransferase (CAT) gene that gives carrier cells resistance chloramphenicol. This gene contained an amber stop codon in the middle that would under normal circumstances render the gene product inactive. The only way the cells would survive this selection is if the tyrosyl-tRNA synthetase aminoacylates the amber suppresser tRNA, thereby enabling the ribosome to create a full-length, functional CAT.

Colonies that were found to survive when 3-(2-naphthyl)-L-alanine was present but not when it was absent were kept because these were presumed to have inserted 3-(2-naphthyl)-L-alanine into the stop codon of CAT. After repeating these steps multiple times, it was found that the synthetase retained its ability to incorporate natural amino acids at the stop codon. In order to increase the specificity toward 3-(2-naphthyl)-L-

alanine, another round of DNA shuffling and two additional rounds of positive selection were performed, which resulted in a significant reduction in the synthetase's affinity toward natural amino acids. The ability of the mutant synthetase to preferentially use 3-(2-naphthyl)-L-alanine as opposed to L-tyrosine was shown when an amber stop codon was inserted into the mouse dihydrofolate reductase (DHFR) gene that was transformed into the synthetase-carrying cells. The gene product was purified using metal affinity chromatography, digested, and analyzed by MALDI mass spectrometry to reveal that 3-(2-naphthyl)-L-alanine was present in the protein fragments.

Part II: Streptavidin specificity

The process involved in predicting the specificity of proteins to different ligands (small organic molecules) remains constant in several regards no matter what protein and ligand is chosen. I have therefore applied the same technique to designing streptavidin mutants that selectively bind to biotin derivatives. Streptavidin is a tetramer, meaning that the functional protein consists of four chains that are not covalently bonded. There are four binding sites on the streptavidin tetramer, each of which is formed by the interaction of two chains. The crystal structure of streptavidin used here shows the atom coordinates of only one chain and one biotin, so this subunit must be duplicated three times and the replicates arranged to form the proper quaternary structure. Since the binding sites are all equivalent, the original biotin molecule is sufficient and the biotin molecules don't need to be replicated. There are several biotin derivatives that have been chemically synthesized and that can be readily procured. These include diaminobiotin, 2-(4'-hydroxyphenylazo)-benzoic acid (HABA), 2,4-Dinitrophenol (DNP), desthiobiotin, and

iminobiotin.[19, 20] The easiest biotin derivative to model is desthiobiotin, because it consists of simply biotin without the sulfur atom. As in other computational designs, the original ligand (in this case biotin) is used to calibrate the parameters of the design.[21] The calibration is done by making biotin the new ligand and changing the settings until the computational algorithm does not mutate the streptavidin scaffold. In this way, biotin is used as a positive control because one would expect the current structure of streptavidin to result in the optimal binding to biotin. Any changes that the computational algorithm introduces would in theory be caused by an error or faulty setting that should be corrected.

Computational Procedures:

The protein that is modified is referred to as a scaffold. The first step in preparing the scaffold is downloading the protein's crystal structure from the RCSB database.[22] This is a text file containing one line per atom. Each atom is marked as being part of an amino acid residue, and in cases where the structure contains multiple protein chains, the chain that the atom is part of. The chains must be merged by removing the chain identifiers and renumbering the residues so that residue identifiers in subsequent chains are greater. There must be a gap of at least two between the residue identifiers of different chains so that the program does not attempt to insert one huge bond joining the ends of the chains, which may be located far apart.

In some instances other types of renumbering may be called for: if a structure contains alternative location characters in the 17^{th} column, these have to be removed and

the residues re-numbered sequentially. Alternative locations are used in order to keep the numbering of residues similar to the numbering found in a homologue protein. However, homology to related proteins is not used in this approach, so the homology can be ignored. The header entries for the residues must be removed because they contain mostly background information on the origin of the structure that is not used by the algorithm in any systematic manner. Any hydrogen atoms existing in the structure are removed and then the whole protein is hydrogenated using the REDUCE program developed by the Richardson lab at Duke University.[23] At this point, the file is converted from the RCSB PDB format to a slightly modified version used as input for the following steps.

All the atoms in the final residue of each chain must be removed except for the nitrogen of the amine group, which will become an oxygen in carboxylic acid group of the penultimate residue. Before the original ligand is appended to the end of the scaffold, it must first be hydrogenated. It is often not practical to reduce the existing ligand at the same time as the protein because the ligand's coordinates might not be present in REDUCE's database of known molecules.[23] To work around this situation, it is possible to generate the complete pdb coordinate file of the original ligand in ChemDraw and Chem3D. These PDB coordinates, which contain the hydrogens, can be oriented the same way as the ligand found in the scaffold and replace that ligand altogether.

Ligands that are covalently bonded to the protein can be treated in approximately the same manner as ligands that are bound by non-covalent interactions. However, there

needs to be a region in 3D space, called subhull, that restricts the position of one or more of the ligand's atoms. Typically, it is the terminal atoms of the ligand next to the scaffold that are immobilized within the subhull.

New ligands are prepared by drawing their structure in the program ChemDraw and optimizing the bond angles and lengths in Chem3D using the MM2 energy minimization algorithm. The resulting structure is saved as a PDB file, which is cleaned up by removing Header and Connect statements and by modifying the spacing of the columns. The contents of the PDB file are copied to another file, where the last column is changed to signify the type of bonding of the atom. This bonding is typically designated as the number 3 for sp3, 2 for sp2, or 0 for hydrogens. A binary representation of the ligand is created and the atoms in the ligand that act as hydrogen bond acceptors or donors must be listed. A hydrogen bond donor is an electronegative atom that is connected to a hydrogen atom. A common example of a hydrogen bond donor is the oxygen atom of a hydroxyl group. A hydrogen bond acceptor, on the other hand, is an electronegative atom that has at least one lone pair of electrons which it partially shares with a hydrogen atom. The distance between a hydrogen bond donor and its hydrogen is typically around than 1.1 Angstroms, while the distance between a hydrogen bond acceptor and the hydrogen that it attracts is 1.6 to 2.0 Angstroms.[24] At least two atoms that are covalently bonded to the electronegative atom in questions must be specified, and the state of the atoms' orbital hybridization (sp2 or sp3) must be listed.

The bond strengths between the atoms that are expected to form hydrogen bonds are specified in a separate file. The atoms that must form hydrogen bonds in order for a design to be kept in the pool of possible solutions must also be provided by the user. This is helpful when it has already been determined that a particular hydrogen bond plays an essential role in maintaining the spatial relationship between the ligand and the protein scaffold. However, the algorithm doesn't know from solely this information that a particular atom in the ligand must hydrogen bond with a particular atom on a residue of the scaffold. Those fine-grain details are specified in another file where the researcher can expressly forbid or promote polar and charged interactions around any atoms. The format of this file is as follows:

```
exclude ASP:*:OD* BT4:*:O* 0.0 5.0
promote ARG:*:HE   BT4:*:O* 1.0 2.5 0.0
```

The first word indicates whether the interaction is encouraged or discouraged. The type of residue is listed next, followed by an asterisk that takes the place of the specific residue number. Next is the name of the atoms, and here again the researcher can use a wildcard to denote multiple atoms. In the first line, OD* means both of the oxygens connected to the Delta carbon of aspartic acid. The next indicator is the name of the ligand, followed by its numeric identifier and the name of the atom. The numbers that follow correspond to the energies that are subtracted or added as a bonus due to demanded hydrogen bonds. The ability to incorporate wildcards gives the algorithm tremendous flexibility and specificity: the researcher can apply a rule to hundreds of residues and thousands of atoms or only to one in particular.

Atoms in the ligand that should not be in direct contact with water molecules, which are assumed to make up the surrounding environment, are listed in a desolvation file. Typically, these are carbons or other non-hydrogen atoms that are part of an aromatic structure. The desolvation file also allows the user to specify the minimum distance between the atoms listed and a water molecule as well as the energy penalty incurred in cases where the distance is smaller than that minimum. The electrostatics interactions, including aromatic rings that may stack upon each other, are listed in the electrostatics file. Here it is possible to specify the charge of any atom in the ligand and the pKA of that atom. Aromatic rings are specified one per line, with every atom within the ring listed along with the atom(s) that it is bound to. Another file describes the surface into which the ligand must fit. The structure of the file is as follows:

```
BT4 * O1  { ARG * HE   1.0 2.5,
        ARG * HH11 1.0 2.5
}
BT4 * N3  { ASP * OD1 1.5 3.5
}
```

The first bracketed segment says that the O1 atom of the biotin ligand named BT4 can accept a hydrogen bond with epsilon hydrogens of all arginines. The following two numbers are the minimum and maximum distance in Angstroms between the two atoms. The second bracketed segment specifies that any of the hydrogens connected to the third Nitrogen atom of the ligand can be donated to the first delta oxygen of aspartic acid, and that the distance between N3 and OD1 must range between 1.5 and 3.5 Angstroms.

After the ligand and scaffold files have been prepared, a new directory called the design directory is created. The global settings that control the process of running the

different steps of the algorithm are listed here. Dead End Elimination (DEE) algorithms are employed to solve combinatorial problems that arise from making minute rotations and translations of the ligand, which are called poses.[25] Each of these poses is evaluated in the context of how well they fit within the existing scaffold. Subsequently, the residues in the scaffold are systematically changed in order to arrive at the scaffold that best fits the particular pose. This determination of how well a particular scaffold fits a ligand pose is done by evaluating the Global Minimum Energy Conformation (GMEC): the lower the GMEC, the more energetically favorable the interaction and the tighter the binding between the ligand and the scaffold.[26]

Before going into more depth on GMEC calculation and Dead End Elimination, it is important to describe how the protein scaffold is systematically changed. The residues in the protein scaffold are subdivided into three basic groups: Evolving Zone (EZ), Molten Zone (MZ), and the rest. The residues in the Evolving Zone are mutated and moved during the course of the algorithm, while side chains in the Molten Zone are only allowed to move but not mutate. This follows from the assumption that residues in the Primary Complementary Surface (PCS) of the enzyme should be allowed to change in order to accommodate a new ligand, but residues that are farther away from the active site only need to be shifted around so as to absorb the shock of mutations in the PCS. The algorithm determines the location of the PCS based on the position of the original ligand, but the user can add or take away the default residues in the Evolving Zone and Molten Zone.

The first step in running the computation is to install the scripts, ligand, and scaffold files inside the design directory. The second step is preparing the scaffold, where the user determines which set of rotamers will be applied in subsequent calculations. A rotamer is a pose of a residue that is rotated by a slight degree. There are three granularities of rotamers that the user can choose from – the densest rotamer set contains a relatively continuous array of rotamers and is best at finding the optimal scaffold conformation. The rotamers of all the residues combined total up to 19,000 in the ultra-dense set. The drawback to using it is that it takes considerably more time to probe rotamers that may be oriented similarly. It is therefore useful to use a rotamer set with light sampling while optimizing other parameters and reserve the ultra-dense set for the final run of the calculation. The global configuration file allows fine-grained control over the construction of the rotamer library. For example, it is possible to use ultra-dense rotamers for just one type of residue (i.e. Phenylalanine) but not for others. This can drastically cut down the computation time in cases where only a few positions are thought to be predicted incorrectly because the proper rotamer was missing from the pool.

When doing a positive control where the new ligand is the same as the original ligand, even the densest rotamer set may not be good enough to recreate the residues in the wild-type scaffold. This can result in bogus predictions where the scaffold is mutated or the rotations of the residues deviate considerably from the wild type. To check whether rotamer density is at fault for a failed positive control, the user is allowed to introduce the wild type rotamers present in the original scaffold into the rotamer set. In fact, he/she

may even choose to build the library with only the wild type rotamers. Provided that the binding of the original ligand to the wild type scaffold is optimal, it would be expected that the wild type rotamers will appear in the scaffold with the lowest GMEC. Another outcome that is almost as good in verifying the validity of the algorithm is if none of the residue types have changed at any position within the scaffold. This result is often the best that can be achieved in many enzymes because they are not meant to bind the ligand very tightly. Instead, enzymes such as aminoacyl-tRNA synthetases bind the ligand transiently and often distort the ligand's internal bond angles and lengths so as to catalyze a particular reaction.[1]

One class of proteins for which positive controls are really useful is antibodies and other proteins whose job is specifically to sequester other molecules with high affinity. Some of the predictions described here have been performed with the protein streptavidin, which although not an antibody, has evolved in bacteria for the purpose of trapping the essential vitamin biotin from the environment.[27]

The next steps in the computation are to generate the ligand ensembles and to construct a three-dimensional docking grid next to the surface of the protein where the ligand ensembles will be fitted. All the residues in the EZ are temporarily mutated into alanines or glycines (configurable at runtime) before the combinatorial search begins. Ligand poses that physically clash with the PCS are removed at the beginning of the calculation. Each ligand pose is taken into consideration one at a time, and the rotamers of different residues are shuffled at different positions within the evolving and molten

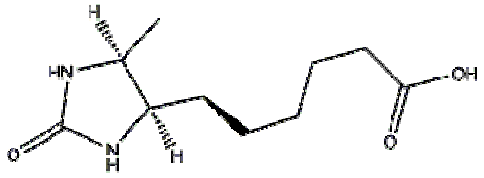
zones. The GMEC is calculated at the end of the shuffling and recorded for later sorting and comparison with other GMECs.

After the combinatorial shuffling, the new scaffold and ligand pose for each rotamer can be viewed as a Kinemage.[28] This graphical representation displays all known bonds (including probable hydrogen or disulfur bonds) present between any two atoms and highlights which residues have been mutated in the scaffold in that particular GMEC. Viewing the structures of each GMEC is useful in determining whether all the molecules are placed in a reasonable manner, but it is impractical to traverse the thousands of GMECs that are generated by the combinatorial search (10,000 GMECs are typically created).

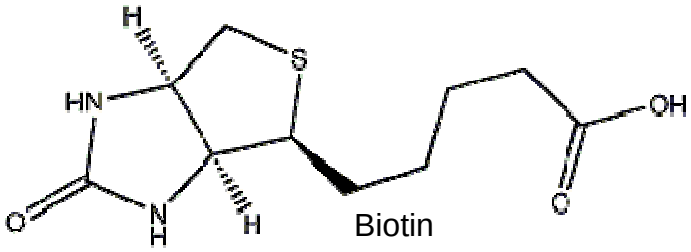
GMEC calculations are segregated into separate jobs that can be run in parallel on multiple nodes on a Linux cluster or on different processors of the same node. The output of each job is sent back to the main node where it is collated. The amount of output can be so large that it overwhelms the network and causes the job queuing mechanism to crash. Consequently, the debug messages generated by jobs are only transferred back to the main node if there appears to be a problem in the calculation. At the end of a run, there are multiple GMEC directories, each corresponding to one a ligand/scaffold combination. All the GMECs are ranked in terms of their energy and heat capacity. The best designs are typically the ones with the lowest GMEC, but the scaffold structure needs to be visually verified to ensure that the expected hydrogen bonds are formed and that no clashes are apparent.

Results:

The complex of streptavidin with biotin has been crystallized and a 3D structure with a resolution of 2.6 Angstroms is available (RCSB ID: 1STP).[29] In this report, the 1STP structure has been used as a starting point for determining mutations that will allow multiple biotin derivatives to be preferentially incorporated. Desthiobiotin is a metabolic precursor of biotin that is the same as biotin all regards except for a missing sulfur atom.



Desthiobiotin



Biotin

The predicted mutations that would allow streptavidin to better bind desthiobiotin are: Serine 45 to Phenylalanine, Tryptophan 79 to Valine, Threonine 90 unchanged, Tryptophan 108 to Serine, and Leucine 110 to Arginine. These predicted mutations will have to be validated experimentally to determine the affinity of the mutated streptavidin to desthiobiotin.

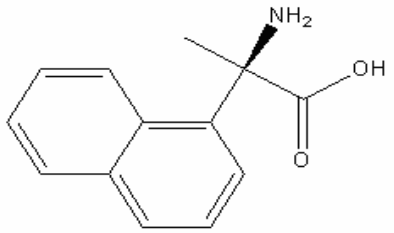
The binding of streptavidin to desthiobiotin has been reported to be several orders of magnitude lower than to biotin[20], although the exact difference depends greatly on the pH in which the assays are performed. The fact that binding of desthiobiotin to wild type streptavidin is reversible has led to its use in agarose purification columns, where a common technique for releasing the bound substrate is to elute the column with buffer containing an increasing concentration of biotin.[30] It would be advantageous, however,

to be able to use both desthiobiotin and biotin for applications that require irreversible binding, such as Enzyme-Linked ImmunoSorbent Assay (ELISA).[31]

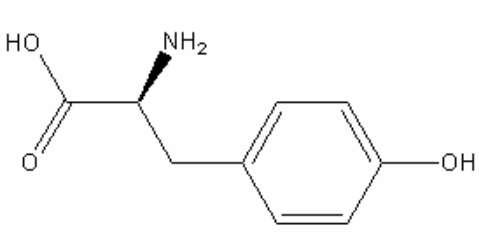
In order to determine how well the algorithm works, biotin was used as a positive control to ensure that the scaffold remained unchanged. Out of the 16 residues in the molten zone (23, 25, 27, 43, 45, 47, 49, 79, 86, 88, 90, 92, 108, 110, 128 and 720), only Serine 45 changed rotamers. In this particular run, no residues were listed in the evolving zone, so they were not allowed to mutate. However, different rotamers of the same residue were allowed, and the fact that only one rotamer changed from the wild type implies that the electrostatic parameters were fairly accurate.

The structure of the Tyrosyl-tRNA synthetase from the mesophilic Archaea *Methanococcus jannaschii* (RCSB ID: 1J1U) has been determined to a 1.95 Angstrom resolution.[32] This structure was used as a scaffold in order to determine what mutations would cause an unnatural analog of L-tyrosine, 3-(2-naphthyl)-L-alanine, to be preferentially accepted into the enzyme's active site and joined to the tRNA. This particular analog of tyrosine was chosen because it is available commercially from Sigma (CAS Number: 58438-03-2), because it's R group is substantially bulkier than tyrosine's thereby changing steric interactions, because the loss of an oxygen could validate the algorithm's ability to take hydrogen bonds into consideration, and because a previous study had shown that it is possible to arrive at a mutant synthetase specific to this analog.[18] The previous study however approached the task of creating the mutant

synthetase through directed evolution, while in this paper a mixed computation and experimental validation approach is detailed.



3-(2-naphthyl)-L-alanine



L-tyrosine

The ligand was prepared in the same manner as biotin, and the algorithm chose twelve residues to be included in the evolving zone based on the position of the original ligand (L-tyrosine): 32, 33, 65, 67, 109, 114, 164, 155, 158, 159, 164, and 177. Of these, residues Ile 33, Leu 65, Ala 67, and Ile 159, and Leu 162 were not mutated by the algorithm. The Gln 109 to Met mutation was ignored because it resulted in the loss of a critical hydrogen bond. The mutation Tyr 114 to Phe was not tested in lab because those residues did not interact with the R group of the ligand. The remainder of the predicted mutations, Tyr 32 to Ala, Asp 158 to Gly, His 177 to Met, Val 164 to Met, and Gln 155 to Ser were introduced into the gene and expressed in recombinant *E. coli*. The wild type tyrosyl-tRNA synthetase was extracted from genomic DNA of *Methanococcus jannaschii*. The tyrosyl-tRNA synthetase gene (NCBI code MJ0389) was PCR amplified from the genomic DNA and introduced into a Stratagene QuickChange plasmid for site-directed mutagenesis.

Experimental Procedures:

The *Methanococcus jannaschii* tyrosyl-tRNA polymerase was amplified by PCR using TAQ polymerase to ensure that the genomic DNA was not degraded. Once that fact was established, the gene was amplified using Pfu to ensure high fidelity replication before cloning. The synthetase gene was then purified using the Qiaquik PCR purification kit, and inserted into pET-14b vector from Qiagen, which was gel-purified and treated with shrimp alkaline phosphatase in order to prevent self ligation. The vector was transformed into *E. coli* XL-10 Gold ultracompetent cells from Stratagene. Colony PCR of the cells that grew was performed to ensure that the plasmid contained the synthetase gene insert, and the gene was sequenced to ensure that no mutations had occurred. *Methanococcus jannaschii* tyrosyl tRNA was inserted into a PSP64 Poly(A) vector from Promega and gel purified from an 8% acrylamide gel. It was expressed in a PQE-80L vector from Qiagen under the control of T7 RNA polymerase.

Introduction of the mutations predicted by the computational algorithm were performed using the QuikChange mutagenesis kit from Qiagen. The primers containing the mutations are as follows:

Y32A

5'-GTTTTAAAAAAAGATGAAAAATCTGCTGCCATAGGTTTTGAACCAAGTGG-3'

H177M

5'-GAGGGATGGAGCAGAGAAAAATAATGATGTTAGCAAGGGAGCTTTTACC-3'

Combined Q155S-3, D158G-1, and V164M-1:

5'-CCAATAATGTCGGTTAATGGTATTCATTATTTAGGCATGGATGTTGCAGTTGGAGGG-3'

The red nucleotides are mutated, while the blue residues are part of the changed codon, but are not mutated. The final primer contained three mutations and was slightly longer, requiring an increase in temperature to 80 degrees C during the denaturation stage of the PCR.

The primers containing the mutations from Schultz et. al. are as follows:

Y32L

5'-GAGGTTTTAAAAAAAGATGAAAAATCTGCTTTAATAGGTTTTGAACCAAGTGGTAAAATAC-3'
5'-GAGGTTTTAAAAAAAGATGAAAAATCTGCTTACATAGGTTTTGAACCAAGTGGTAAAATAC-3'

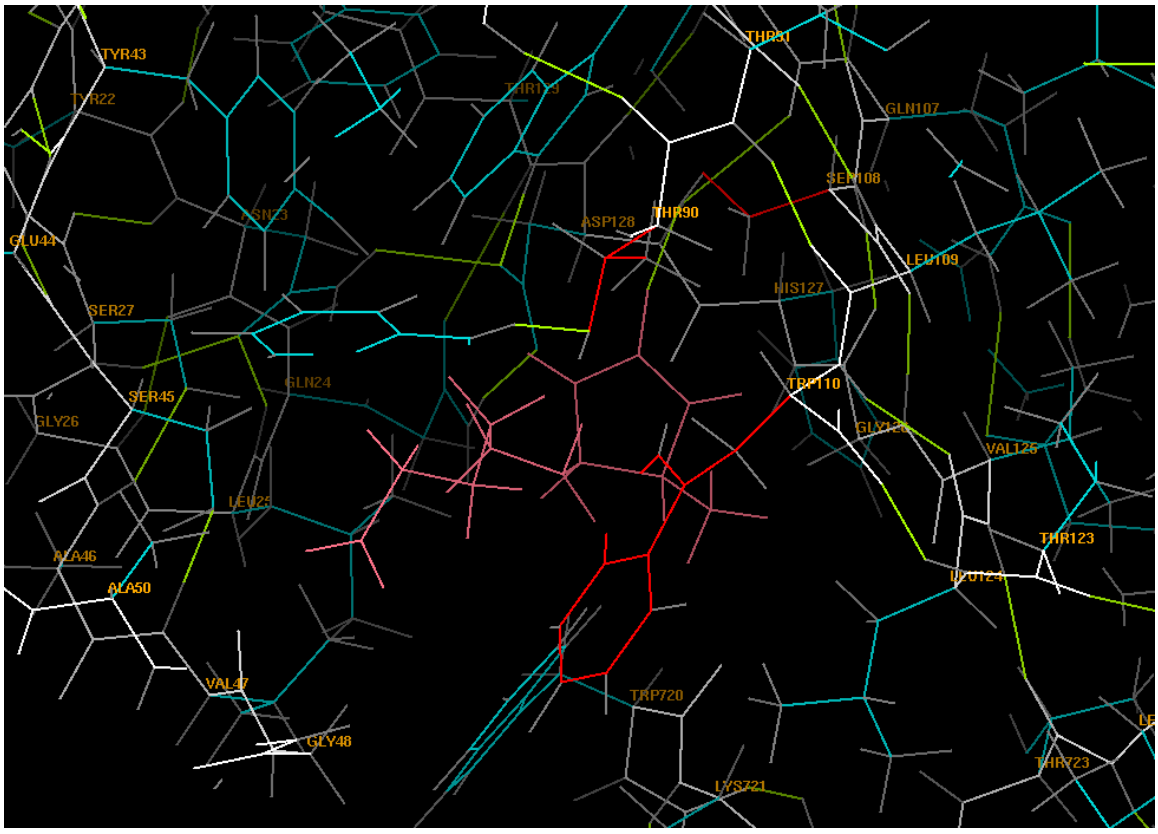
Combined D158P-3, I159A-3, L162Q-1, A167V-3

5'-GCAGGTTAATCCGGCGCATTATCAAGGCGTTGATGTTGTGGTTGGAGGGATGGAGCATGGAGC-3'

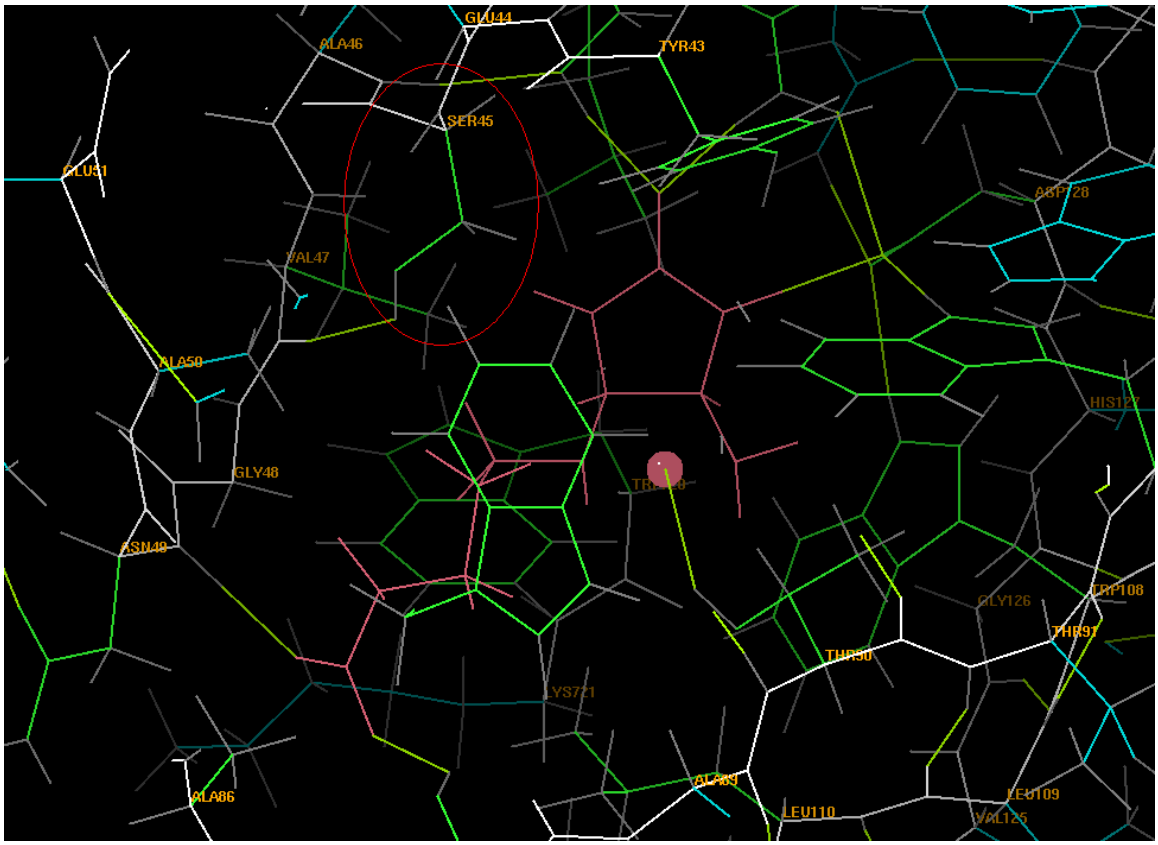
After mutating the wild type tyrosyl-tRNA synthetase according to the computational prediction and according to the previous study by Schultz et. al. [18], the genes were transferred to a pAK plasmid and expressed in a BL21-DE3 *E. coli* strain that is auxotrophic for tyrosine. This was achieved by knocking out the trpC gene and verifying that this strain does not grow on minimal media where tryptophan was not provided. This strain was grown in defined minimal media containing increasing concentrations of 3-(2-naphthyl)-L-alanine. Growth of cells in this media has been slow and capricious, since colonies that grow on agar plates do not grow in liquid media containing the same ratio of 3-(2-naphthyl)-L-alanine to tyrosine. It remains to be seen

whether this is due to the toxic effects of the tyrosine analog. The process of evolving the strain to withstand this and other unnatural amino acids is ongoing, as is the investigation into the effectiveness of the mutant synthetases at aminoacylating suppressor tRNAs.

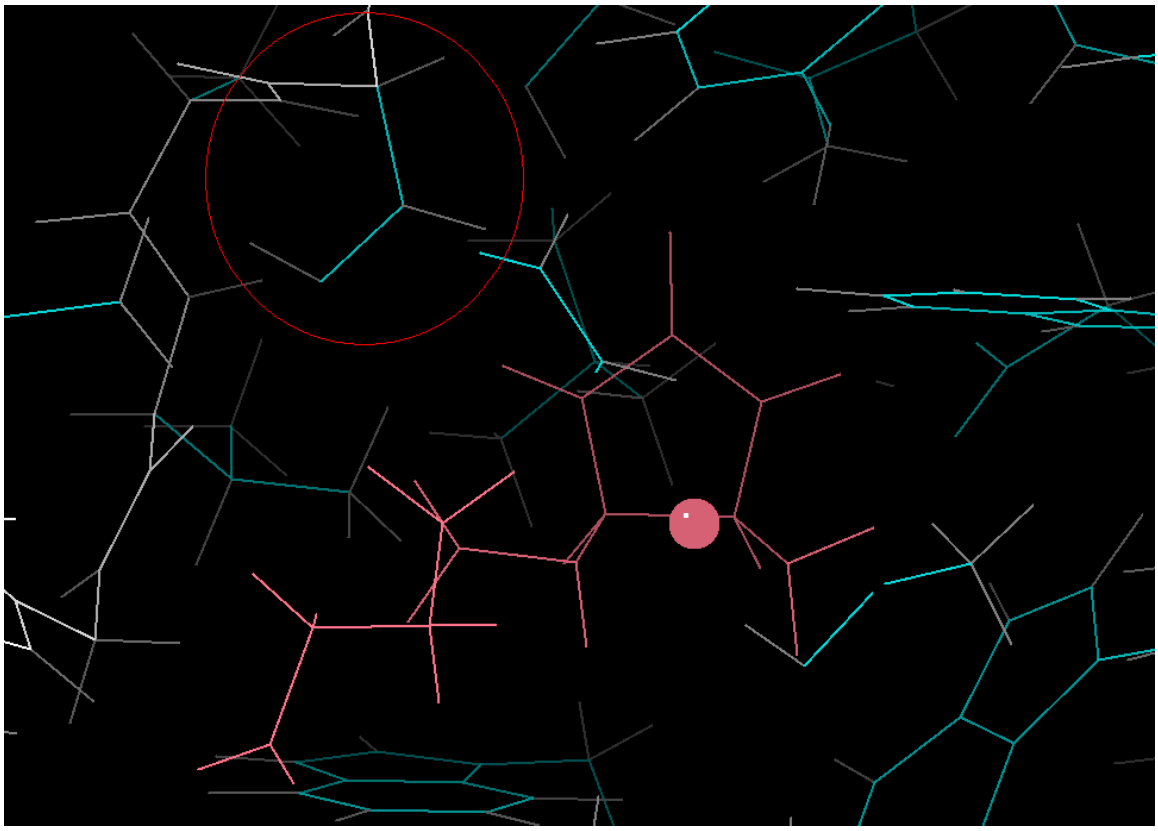
So far, the best computational design for how to mutate Streptavidin to accept desthiobiotin has been Arginine 53 to Valine, Tryptophan 108 to Serine, and Leucine 110 to Tryptophan. The other residues present in the evolving zone that were not mutated by the algorithm were Phenylalanine 29, and Threonine 90. The proposed structure of the mutant enzyme can be seen below:



The residues highlighted in red are different from the wild type, while the green lines signify hydrogen bonds. This can be contrasted to the design in which biotin was used as the new ligand:



Please note the red circle surrounding shows the rotamer of Serine 45 - this has changed compared to the original crystal structure of streptavidin with biotin (below).



Even though the rotamers in the two structures are technically different, their orientation and position relative to the ligand are very similar, increasing the likelihood that the variation is insignificant and that the parameters used in the computation are applicable to real-world situations.

Conclusion:

In order to determine whether Schultz et. al's mutant tyrosyl-tRNA synthetase works and to compare its efficiency with the mutations given by the computational algorithm, I reproduced Schultz et. al's mutations in this study. The mutations revealed by Schultz et. al were Tyr 32 to Leu, Asp 158 to Pro, Ile 159 to Ala, Leu 162 to Gln, and Ala 167 to Val. The computational method described here has been shown to be successful in positive control simulations where existing ligands are treated as new

molecules that need to be fitted into enzymes. These controls are facilitating the determination of energetic parameters that can be used for fitting modified ligands.

The settings vary from protein to protein because the interactions of scaffolds to their ligands are not all the same: in many enzymes, evolution has favored weak binding because the product must be released. Positive controls are not as easy to verify for weak-binding proteins because the mutations predicted by the computational algorithm may actually provide a tighter interaction and lead to a lowered catalytic ability. The experimental results are promising for simulations where analogs of biotin and tyrosine are fitted into streptavidin and tyrosyl-tRNA synthetase, respectively. More tests need to be performed to reliably quantify the strength of binding and catalytic activity of these enzymes. It is expected that this in-silico approach will work equally well for other classes of proteins.

Acknowledgements:

I would like to thank Dr. Andrew Ellington, Randy Hughes, Dr. Matt Levy, Casey Cole, and Micheleen Harris from the Ellington Lab at the University of Texas at Austin as well as Dr. Homme Hellinga, Dr. Katarina Midelfort and Dr. Loren Looger from the Hellinga Lab at Duke University for their invaluable help throughout this project.

References:

[1] S. W. Lee, B. H. Cho, S. G. Park and S. Kim, *J Cell Sci* **2004**, *117*, 3725-3734.
[2] A. Marintchev and G. Wagner, *Q Rev Biophys* **2004**, *37*, 197-284.
[3] J. M. Ogle and V. Ramakrishnan, *Annu Rev Biochem* **2005**, *74*, 129-177.
[4] P. F. Agris, F. A. Vendeix and W. D. Graham, *J Mol Biol* **2007**, *366*, 1-13.

[5] Y. Nakamura and K. Ito, *Genes Cells* **1998**, *3*, 265-278.
[6] H. Beier and M. Grimm, *Nucleic Acids Res* **2001**, *29*, 4767-4782.
[7] L. Wang, A. Brock, B. Herberich and P. G. Schultz, *Science* **2001**, *292*, 498-500.
[8] S. J. Anthony-Cahill and T. J. Magliery, *Curr Pharm Biotechnol* **2002**, *3*, 299-315.
[9] C. Grangeasse, A. J. Cozzone, J. Deutscher and I. Mijakovic, *Trends Biochem Sci* **2007**, *32*, 86-94.
[10] C. M. Spickett, A. R. Pitt, N. Morrice and W. Kolch, *Biochim Biophys Acta* **2006**, *1764*, 1823-1841.
[11] C. W. Wilkinson, *Peptides* **2006**, *27*, 453-471.
[12] R. A. Mehl, J. C. Anderson, S. W. Santoro, L. Wang, A. B. Martin, D. S. King, D. M. Horn and P. G. Schultz, *J Am Chem Soc* **2003**, *125*, 935-939.
[13] J. Xie and P. G. Schultz, *Curr Opin Chem Biol* **2005**, *9*, 548-554.
[14] H. Yamanishi and T. Yonesaki, *Genetics* **2005**, *171*, 419-425.
[15] J. C. Anderson and P. G. Schultz, *Biochemistry* **2003**, *42*, 9598-9608.
[16] L. Wang and P. G. Schultz, *Chem Biol* **2001**, *8*, 883-890.
[17] J. C. Anderson, N. Wu, S. W. Santoro, V. Lakshman, D. S. King and P. G. Schultz, *Proc Natl Acad Sci U S A* **2004**, *101*, 7566-7571.
[18] L. Wang, A. Brock and P. G. Schultz, *J Am Chem Soc* **2002**, *124*, 1836-1837.
[19] O. Livnah, E. A. Bayer, M. Wilchek and J. L. Sussman, *FEBS Lett* **1993**, *328*, 165-168.
[20] J. D. Hirsch, L. Eslamizar, B. J. Filanoski, N. Malekzadeh, R. P. Haugland, J. M. Beechem and R. P. Haugland, *Anal Biochem* **2002**, *308*, 343-357.
[21] O. H. Laitinen, V. P. Hytonen, H. R. Nordlund and M. S. Kulomaa, *Cell Mol Life Sci* **2006**, *63*, 2992-3017.
[22] H. Berman, K. Henrick, H. Nakamura and J. L. Markley, *Nucleic Acids Res* **2007**, *35*, D301-303.
[23] J. M. Word, S. C. Lovell, J. S. Richardson and D. C. Richardson, *J Mol Biol* **1999**, *285*, 1735-1747.
[24] T. K. Harris and A. S. Mildvan, *Proteins* **1999**, *35*, 275-282.
[25] L. L. Looger and H. W. Hellinga, *J Mol Biol* **2001**, *307*, 429-445.
[26] I. Lasters, M. De Maeyer and J. Desmet, *Protein Eng* **1995**, *8*, 815-822.
[27] M. Seki, *Med Res Rev* **2006**, *26*, 434-482.
[28] D. C. Richardson and J. S. Richardson, *Protein Sci* **1992**, *1*, 3-9.
[29] P. C. Weber, D. H. Ohlendorf, J. J. Wendoloski and F. R. Salemme, *Science* **1989**, *243*, 85-88.
[30] B. Kuhn and P. A. Kollman, *J Med Chem* **2000**, *43*, 3786-3791.
[31] J. Gross, R. Moller, W. Henke and W. Hoesel, *J Immunol Methods* **2006**, *313*, 176-182.
[32] T. Kobayashi, O. Nureki, R. Ishitani, A. Yaremchuk, M. Tukalo, S. Cusack, K. Sakamoto and S. Yokoyama, *Nat Struct Biol* **2003**, *10*, 425-432.